

GARUDA USER MANUAL

Version 2.0

Compiled & authored by : Sukeshini & SOA team
Reviewed by : R.Sridharan

SOA Group
Centre for Development of Advanced Computing
Knowledge Park, Bangalore
(C-DAC KP)

PREFACE

GARUDA User's Manual focuses on the usage of GARUDA Grid computing systems. The document presents the overview of GARUDA network, its architecture and other GARUDA Components. GARUDA User's Guide is intended for Application Developers, researchers and end-users of the GARUDA Grid.

Users are required to refer product manuals for the installation procedure of individual GARUDA components. GARUDA site administration aspects are not covered in this document.

The first three sections provide a brief overview of GARUDA network, its architecture with appropriate illustration. The fourth section introduces the GARUDA Access Portal (GAP), - an in-house job submission interface. It includes directives about invoking the portal, registration, security-system used and types of Job submission.

The fifth section deals with Paryavekshanam – Grid monitoring tool developed in-house. It used to monitor various resources of the GARUDA Grid, the users can troubleshoot the problem related to GARUDA Grid.

The sixth section is related to Problem reporting and Request Tracking used by GARUDA for follow up of the problems reported. FAQ's, Glossary and references are provided at the end of the document.

All the links mentioned in this document are also accessible from <http://192.168.60.40>

TABLE OF CONTENTS

| | |
|---|-----------|
| 1.0 GARUDA OVERVIEW | 6 |
| 2.0 GARUDA ARCHITECTURE | 8 |
| 2.1 GARUDA Connectivity..... | 8 |
| 2.2 GARUDA Components..... | 9 |
| 2.3 IGCA | 11 |
| 2.4 VOMS..... | 18 |
| 3.0 GARUDA USAGE FRAMEWORK | 20 |
| 4.0 GARUDA ACCESS PORTAL (GAP).. | 22 |
| 4.1 Pre-requisite for using GARUDA Access Portal (GAP)..... | 22 |
| 4.2 Accessing GARUDA Access Portal (GAP) | 23 |
| 4.3 Job Management | 24 |
| 4.4 Resources | 28 |
| 4.5 File Browser..... | 30 |
| 4.6 Accounting | 31 |
| 4.7 MyProxy..... | 31 |
| 4.8 Virtual Organization Management (VOMS) Proxy | 33 |
| 4.9 Partner's Site | 34 |
| 4.10 Notices | 34 |
| 5.0 DATA MANAGEMENT | 35 |
| 5.1 Data Grid Solution..... | 35 |
| 5.2 GSRM Accessibility in GARUDA..... | 36 |
| 5.3 Pre-requisites for using GSRM services..... | 39 |
| 5.4 GSRM Client Interfaces | 40 |
| 6.0 PARYAVEKSHANAM | 42 |
| 6.1 Pre-requisite for using Paryavekshanam | 43 |

| | |
|--|-----------|
| 6.2 Accessing Paryavekshanam..... | 43 |
| 6.3 Features..... | 43 |
| 6.4 Web Pages | 43 |
| 7.0 COMPILER SERVICE..... | 47 |
| 7.1 Pre-requisite for using SOA Compiler | 47 |
| 7.2 Accessing SOA Compiler | 47 |
| 7.3 Compiler GUI..... | 47 |
| 7.4 Makefile Generation | 49 |
| 7.5 Building the Application..... | 49 |
| 7.6 Accessing Compiler Service..... | 49 |
| 8.0 UTILITY SERVICES..... | 51 |
| 8.1 Login Web Service | 51 |
| 8.2 Accounting Web Service | 54 |
| 9.0 REQUEST TRACKER..... | 56 |
| 10.0 FREQUENTLY ASKED QUESTIONS (FAQ's) | 58 |
| 11.0 LIST OF REGIONAL AUTHORITIES (RA's)..... | 62 |
| 12.0 REFERENCES..... | 65 |
| 13.0 GLOSSARY | 66 |

LIST OF FIGURES

- Figure 1: GARUDA-NKN connectivity
- Figure 2: User Registration Flowchart
- Figure 3: GARUDA Usage Framework
- Figure 4: GARUDA Architecture
- Figure 5: GARUDA Access Portal Home page
- Figure 6 : Basic job submission
- Figure 7: Advanced job submission
- Figure 8: Job Information
- Figure 9: Resource Reservation
- Figure 10: File Browser
- Figure 11: Accounting
- Figure 12: MyProxy
- Figure 13: VOMS Proxy
- Figure 14: GSRM Usage by GARUDA Users
- Figure 15: Home page of Paryavekshanam
- Figure 16: Grid Overview Page
- Figure 17: Network Overview Page
- Figure 18: Alert Page
- Figure 19: Compiler GUI
- Figure 20: Raising a Issue ticket in GARUDA Request Tracker
-
- Table 1: List of DPM 'C' APIs

1. GARUDA OVERVIEW

GARUDA is a collaborative environment of scientific researchers and experimenters on a nation wide grid of computational nodes, mass storage and scientific instruments that aims to provide the technological advances required to enable data and compute intensive science for the 21st century.

GARUDA aims at strengthening and advancing scientific and technological excellence in the area of Grid and Peer-to-Peer technologies. The strategic objectives of GARUDA are to:

- Create a test bed for the research & engineering of technologies, architectures, standards and applications in Grid Computing.
- Bring together all potential research, development and user groups to develop a national initiative on Grid Computing.
- Create the foundation for the next generation grids by addressing long term research issues in Grid Computing.
- Enable users to have seamless access to the super computing facilities available at various institutes to run their application, which have high performance computing/ communication requirements.

The Department of Information Technology (DIT), Government of India has funded the Centre for Development of Advanced Computing (C-DAC) to deploy the nation-wide computational grid 'GARUDA' connecting 17 cities across the country in its *Proof of Concept (PoC) phase*, ended on March 2008, accomplished its deliverables by connecting 17 cities across 45 academic and research institutes country wide along with the required softwares for managing grid computing

applications.

The *Foundation phase* (April 2008 – August 2009) successfully aimed at developing Service oriented Architecture (SOA) framework, improving network stability and upgrading grid resources.

The 36 months *Operation phase* promises primarily on the grid enablement of applications of national priority viz Disaster Management (DMSAR), Bio informatics.

GARUDA will accelerate India's drive to turn its substantial research investment into tangible economic benefits.

2. GARUDA ARCHITECTURE

2.1 GARUDA Connectivity

National Knowledge Network (NKN) is a multi 10 Gbps network of High Speed Computing & Communication system, implemented by the Department of Information Technology (DIT). The ultimate aim of NKN is to unite stakeholders in science, technology, higher education, R&D and governance using network speeds of tens of gigabits per second coupled with extremely low latencies. NKN encourages collaboration and the creation of new national intellectual assets, enabling the sharing of high-performance computing facilities, e-libraries, virtual classrooms, and more.

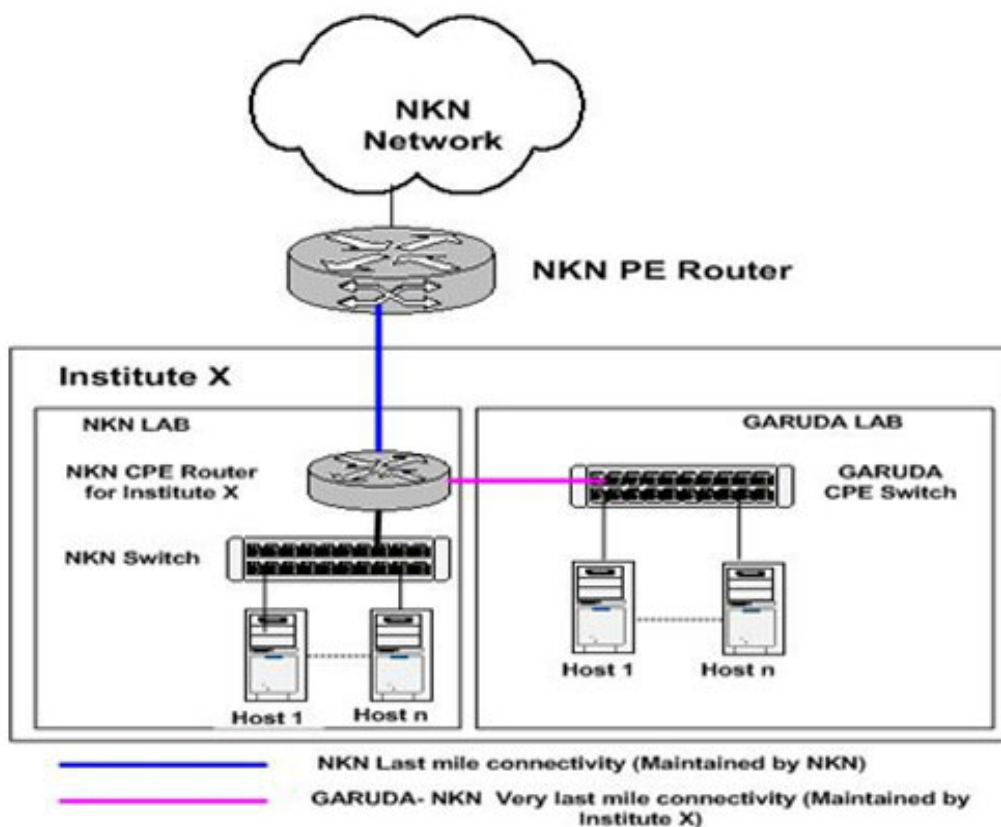


Figure 1: GARUDA-NKN Connectivity

GARUDA Grid network is entirely migrating to National Knowledge Network (NKN), the facility built by Government of India. The NKN's design philosophy is to build a scalable network which can expand both in terms of accessibility and speed. The last mile connectivity to the participating agencies varies from 1 Gigabits/sec, 100 Megabits/sec to 10 Megabits/sec.

NKN features includes:

- High capacity and scalable backbone,
- Highly reliable and available,
- Supports strict QoS and security,
- Have wide geographical coverage and
- Has common standard platform.

2.2 GARUDA Components

2.2.1 Access Methods

Currently, access to the Grid resources is through the high-speed communication fabric. Access through satellite based communication channels is also being explored as part of a research initiative to integrate the GARUDA terrestrial grid with a satellite based grid.

- **GARUDA Portal** : GARUDA portal is used to access and execute Grid applications from a conventional Web browser and other desk top tools. It hides the complexity of the GRID from the users.
- **Video Conferencing**: Access through Video Conferencing is also available. Multi point video conferencing enable people to interact simultaneously from different place and supports their collaborative environment.

2.2.2 Data Management

To enable data oriented applications, GARUDA provides an integrated and distributed

data storage architecture by deploying the Storage Resource Broker (SRB) from Nirvana. SRB creates and maintains a Global Namespace across multiple heterogeneous and distributed storage systems in the Grid so that users can access their data from any where at any time.

2.2.3 Collaboration Environment

GARUDA initiated virtual communities will focus on collaborative projects in specific technology/application domain. The virtual communities are supported by forming Virtual Organizations to look into the collaborative Project Development across GARUDA Partners. Initially the collaborations will focus on Grid tools & Services, High-speed Networking and Information Security in Technology domain while Earth sciences, Life Sciences, Material Sciences and High Energy Physics are identified in the Application domain.

2.2.4 Management and Monitoring Tools

Grid monitoring and management centre at C-DAC, Bangalore helps in managing and monitoring all the components in the GARUDA Grid. Components like Network, Compute nodes, Storages, and Software correctness are monitored on a 24X7 basis. State-of-the-art display walls and advanced software like **Paryavekshanam** developed at C-DAC, help in effectively monitoring the health and utilization of various components of the Grid.

2.2.5 Resources

In this collaborative grid project, various resources such as high performance computing systems (HPC) and satellite based communication systems have been committed by different centers of C-DAC and the partners of GARUDA. It may be noted that the resources being diverse in nature, one of the major challenges of GARUDA is to deploy appropriate tools and middleware to enable applications to run seamlessly across the grid.

2.2.6 Middleware and Grid Security

GARUDA Grid deploys Globus Toolkit, version 4.0.7 (GT4), for the operational middleware functionality. The resource management and scheduling in GARUDA is based on the deployment of industry grade schedulers in a hierarchical architecture. At the cluster level, scheduling is achieved through Load Leveler for AIX platforms and Torque for Solaris and Linux clusters. At the Grid level, Gridway meta-scheduler enables reliable and efficient sharing of computing resources managed by different LRM (Local Resource Management) systems.

SOA separates individual Grid functionalities into distinct Grid Services distributed over a network and can be combined and reused to create scientific and business applications. The SOA based GARUDA architecture deploys GSI (Grid Security Infrastructure), Virtual Organization (VO) Management and MyProxy mechanisms to enforces the security methods of the GARUDA Grid.

A central concept in GSI authentication is the certificate. Every user and service on the Grid is identified via a certificate. The certificates are issued and signed by a trusted Certificate Authority (CA) in a Grid.

Introducing Virtual Organizations (VO), Grid users are partitioned into different domains thereby introducing another level of authentication from various local VO servers. *MyProxy*, a credential management service provides functionalities to manage X.509 Public Key Infrastructure (PKI) security credentials. Each credential is protected by a password that is provided by the user at the time of storage. The credential can be retrieved later from the repository for use.

Indian Grid Certificate Authority (IGCA) accredited by IGTF, acts as certificate authority and will issue X.509 security having the Grid credentials. Indian researchers can now request user and host certificates to IGCA and get access to worldwide Grids.

2.3 Indian Grid Certificate Authority (IGCA) – Certificate Requests

Requesting for the certificates, indicate that you accept the Certificate Policy and Certification Practice Statement (CP/CPS) and that you agree to the subscriber Obligations specified in that document.

Download the IGCA User/ Host Certificate Application form from the website.

<http://ca.garudaindia.in/index.php/certificate/>

New Grid users

Following are the steps for novice Grid users:

1. ***To apply for IGCA User Certificates***, request for the User Certificate on-line.
 - Create a *Certificate Signing Request (CSR) online*. Follow *User -> Request a Certificate* and click *Request a User Certificate*.
 - Make sure the request submission should be on the computer where you want to store your public/private keys.
 - Note down the unique **serial no.** generated and enter in the application form provided in the *CSR no.* field.

Advanced Grid users

2. ***To apply for IGCA Host Certificates***, request for the Host Certificate on-line.
 - Generate the *Key Pair* locally. Pre-requisite: Openssl should be installed in your local machine and the openssl commands are included in the path.
 - Follow either of the Options for requesting the certificates

Option -1

- Download the file IGCA_Host_Req.tar.gz to a Unix-like machine and decompress this tar file.
- Give execute permission to the bash file IGCA_Host_Req.sh.
- On executing the bash file, you will be prompted for the FQDN (Fully Qualified Domain Name) of the host for which certificate being requested.
- Upon completing the host certificate request procedure, two files *<FQDN>.hostreq.pem* (host certificate request) and *<FQDN>.hostkey.pem* (private key for your certificate request) are generated.
- Copy the *<FQDN>.hostkey.pem* into */etc/grid-security/* directory of the server for which you requested the certificate as *hostkey.pem*, and change the permissions.
- Upload the *<FQDN>.hostreq.pem* to IGCA. Click Request Host Certificate and GoTo User -> Request a Certificate and Click the Request a Host certificate.

Option -2

- On any Linux flavor with *openssl* version greater than 0.9.7a, run the command, with the parameters of FQDN replaced with your FQDN HostName.

```
[admin@grid]# openssl req -new -days 365 -sha1 -newkey rsa:1024 -nodes -  
keyout      hostkey.pem      -out      hostcert_request.pem      -subj  
'/DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSFCN=FQDN'
```
- Make sure the Private Keys Generated is used only for the generated Host.
- It generates 2 files (*hostkey.pem* - Private Key File and *hostcert_request.pem* - Public Key File).
- Copy the *hostkey.pem* into */etc/grid-security/* directory of the server for which you requested the certificate and change the permissions.

- Then go to the *Request Host Certificate* and upload the *hostcert_request.pem* file and with details same as in the Host Application Form and confirm your Request. Follow User -> Request a Certificate and then Click Request a Host Certificate.

Note down the unique serial no. generated and enter in the application form provided in the CSR no. field.

3. Setup an interview with a local Registration Authority (RA) along with user certificate application form and Photo ID (Passport, PAN Card, Work ID or Driver's License) to verify your identity.

Refer Section 11: List of available RA's

4. Complete the application form and submit the scanned copy to igca@cdacb.ernet.in or Fax the application form along with the ID Proof to: +91-80-2524 7724
5. **CRIN pin is sent only for USER CERTIFICATE.** An automated mail notifying the details for downloading your certificates will be sent once the Certificate is issued. Note your Serial Number and to download your Certificate. You need to store securely **CRIN** pin for further usage (like: Revoking).
6. The procedure for obtaining the host certificate is completed.
7. To download and import the User certificates, follow the links provided in the email, for automatic update of certificate onto the browser or downloading the certificates. Then you need to combine the private and public keys to use the certificate.
8. Export the certificate from your browser.

Copy your certificate and private key to removable media which should be kept in safe. File extension of the exportation will be .pfx if exporting user certificate from Internet Explorer (IE), and .ps12 if exporting it from Firefox.

- Exporting the digital certificate with the private key from **Internet Explorer(IE)**:
 - ➔ Open the IE browser and go to “Tools”-->“Internet Options” --> “Content” tab, then choose “**Certificates**”.
 - ➔ Click your certificate that you want to export. --> then click “**Export**” button.
 - ➔ Click “Next” in the “**Export Wizard**” window and select “**Export private key**”.

Click **“Next”**.

- ➔ Make sure **“Personal Information Exchange -PKCS#12”** is checked, and also the **“Enable strong protection”**.
- ➔ The **“Delete private key if successful”** must be unchecked. The **“Include all certificate in path”** button should be unchecked, too. Click **“Next”**.
- ➔ Type the pass-phrase(twice) to protect your private key. Click **“Next”**.
- ➔ Type the name of the file and where you want to store your certificate. Click **“Next”**--> Click **“Finish”**

➤ Exporting the digital certificate with the private key from **Firefox**:

- ➔ Open the Firefox browser and go to **“Tools”-->“Options”** --> **“Advanced”** --> and then **“Encryption”** Tab.
- ➔ Click **“View Certificates”** --> Goto to **“Your Certificates”** tab and select the Certificate which you want to backup.
- ➔ Enter the Proper Filename and the Path where it needs to be stored.
- ➔ Type the passphrase(twice) that you use to protect your private key. Click **“Ok”**.
“Successfully backedup your security certificate(s) and private key(s)” message will be shown.

➤ Convert the Certificate to PEM format.

In order to use grid resource, you need to split pfx(p12) file into it's two components, your public (usercert.pem) and private (userkey.pem) keys. (Say test.p12 is the file you took backup certificate from the browser.) You need to provide the import password which you had set during the backup certificate from the browser.

➤ **Generating usercert.pem**

```
[test@grid]$ openssl pkcs12 -in test.pfx(p12) -clcerts -nokeys -out usercert.pem
usercert.pem
```

Enter import Password:

MAC verified OK:

➤ **Generating userkey.pem**

```
[test@grid]$ openssl pkcs12 -in test.pfx(p12) -nocerts -out userkey.pem
```

Enter Import Password:

MAC Verified OK

Enter PEM pass phrase:

Verifying Password - Enter PEM pass phrase:

Verified OK

- Then it asks for setting PEM pass phrase, which is used to generate the proxy from your certificates.
- The permissions on your keys should be changed to that only you can read the private key and everyone can read-only the public key:

```
[test@grid]$ chmod 644 usercert.pem
```

```
[test@grid]$ chmod 400 userkey.pem
```

```
[test@grid]$ grid-proxy-init -cert usercert.pem -key userkey.pem
```

Your Identity: /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=henry (emailid@domain.in)

Enter Grid Pass phrase for this identity:

Here you need to provide the pass phrase which you had set while generating the userkey.pem from the above commands. This will generate the proxy of default 12 hours.

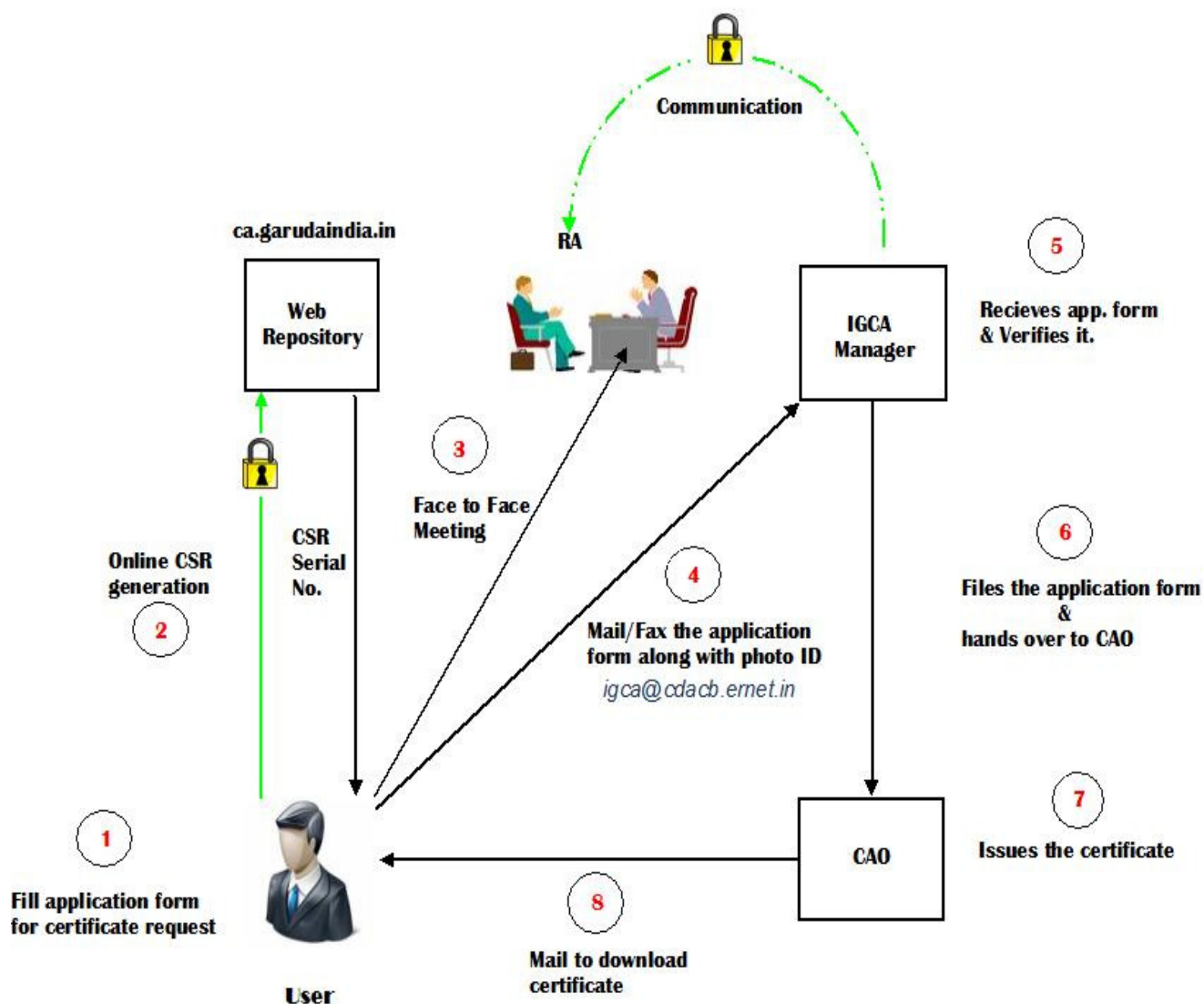


Figure 2: User Registration Flowchart

- For more details about requesting user / host certificates, please go to <http://ca.garudaindia.in/index.php/certificate/certificate-request/>

2.4 Virtual Organization Management System (VOMS)

2.4.1 User registration process to a VO

If you want to become a member of a VO, follow the steps given below:

1. You require a valid user certificate from a CA recognized by VO Server. Create a .p12 certificate from your .pem certificate. Upload the .p12 certificate in your browser.
2. Request for the membership
 - User needs to point to his browser (loaded with his/her certificate) to the following URL: *https://<voms-admin server hostname>:8443/vomses/*
 - User should then select the VO to register into from the list of available VO's.
 - Complete the VO Registration Form and the request will be sent to the VO-Admin of that particular VO by the system.
3. User need to confirmation this request by clicking to the email sent by system with instructions on how to proceed.
4. When VO-Admin approves the request, the user will become a member of that VO with specified roles. (The VO Subscription Page lists the pending user requests.)
5. Notification to the user about approval/rejection of his/her request is sent via email.

2.4.2 List of available VO's in GARUDA

We have identified few VO's. User can get the complete list of VO's at <https://144.16.192.37/vomses/>. User must belong to at least one VO. can be which are listed below:

1. IndianHeritage - This VO is for those who are working in the filed of preserving Indian Heritage through NLP(Natural Language Processing) and other technologies.
2. GarudaAdmin - This is for the System Administrators responsible to different resource sites of GARUDA Grid. All the System Administrators from C-DAC and partners site

are enrolled in this VO.

3. OSDD:- This VO is meant for application members involved in Open Source Drug Discovery activity.
4. Others - A General VO for those who do not fall on any of the above Categories.

Additional VOs can be added to this list, based upon the formation of different virtual communities among the Garuda users". Based on the feedback and the requests of the members of the individual VO's respective administrator will be responsible to assign relevant roles to the members.

3. GARUDA USAGE FRAMEWORK

GARUDA framework enlists its Network, Resources, Federated Information Services, Security with Authentication and Authorization services, Job Management (comprising Data movement, Scheduling, Reservation and Accounting), Access mechanisms like Access Portal, Workflow tools and Problem Solving Environment (PSE's).

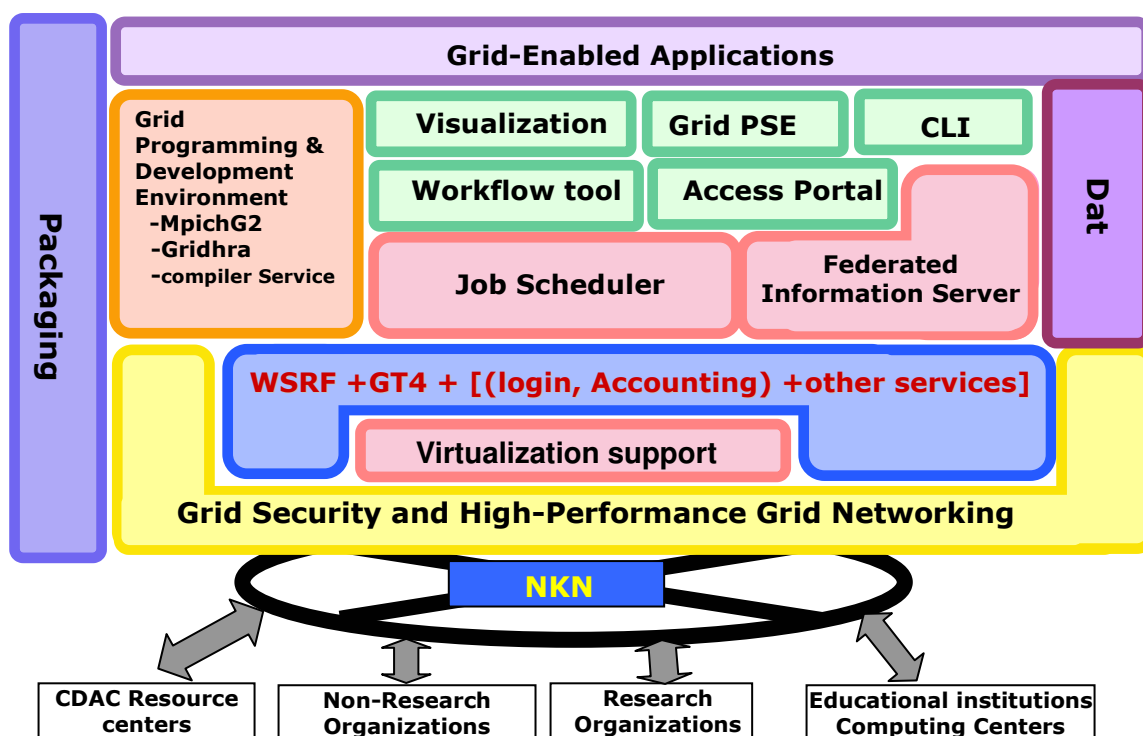


Figure 3: GARUDA Core Components

GARUDA Network entirely depends of National Knowledge Network (NKN) connecting all the compute and storage resources spread through out the nation. Federated Information System helps in keeping track of information about all the GARUDA resources. Globus 4.0.x is an open source toolkit for building computing grid.

GARUDA Access Portal (GAP), Command line interface (CLI), Workflow tools and Problem Solving Environment (PSE's) are various access mechanisms for GARUDA users. Grid users can submit and manage their jobs using web browser based GAP or through CLI. Job scheduler takes care of the actual job submission on the GARUDA resources. Advance reservation of resources allows GARUDA users to run their jobs at specific time slots. Visualization tools enable the Grid users to graphically present their information. PSE provides all the computational facilities to solve a target class of problems allowing users to run them without specialized knowledge of underlying hardware or software.

Virtual Organization Management System (VOMS) manages the Virtual Organization (VO's) in GARUDA Grid and manages security issues. Service Oriented Architecture (SOA) based GARUDA exposes various services like Login, Compiler, Accounting for providing essential facilities for grid users. GARUDA extends virtualization facility to allow multiple operating system to run simultaneously on resource in a safe and efficient manner.

4. GARUDA ACCESS PORTAL (GAP)

A generic Job Submission environment, GAP enables researchers and scientists to execute their applications on Grid from a conventional web browser. Both Sequential and Parallel jobs can be submitted to GARUDA Grid through Portal. It provides a web interface for viewing the resources, and for submitting and monitoring jobs.



Figure 4: GARUDA Portal Home page

4.1 Pre-requisites for using GAP

- Portal users need to set the following in their ~/.bashrc file.

```
export GW_LOCATION=/opt/gw/
```

```
export GLOBUS_LOCATION=/opt/asvija/GLOBUS-4.0.7/  
source /opt/asvija/GLOBUS-4.0.7/etc/globus-user-env.sh  
export PATH=/usr/local/jdk1.6.0_10/bin:          GW_LOCATION/bin:  
/opt/garudaesv/bin:/opt/voms_client/bin:$PATH  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/voms_client/lib:
```

- Users should have login on Grid Head Node(gridfs) and valid Globus Certificates to submit jobs to the grid.
- Users need to enable POP-Ups in their browser.
- New users need to acquire the login and Globus certificates for submitting their jobs to the grid.
- Supported browsers versions

Linux Platform

- Mozilla version 1.6 and above.
- Firefox version 1.0.4 and above.

Windows Platform

- Internet Explorer version 6.0.2 and above.
- Firefox version 1.5.0.1 and above.

Solaris Platform

- Netscape version 6.2.3 and above.

4.2 Accessing GAP

4.2.1 Invoking the Portal

Type <http://192.168.60.40/GridPortal1.3/> (to access the Portal through GARUDA Network) or <http://203.200.36.236/GridPortal1.3> (to access the Portal through Internet) in the address bar of the web browser to invoke the Portal. It is preferable to access the Portal through GARUDA Network, since it is much faster than the Internet.

4.2.2 Login

In order to access the facilities of Grid Portal such as Job Submission, Job Status tracking, Storing(Uploading) of Executables and View Output/Error data, the user has to login into the Portal using the User's Login Form in the Home page of the Portal.

a) New users are required to click *Sign up* in the User Login Form, which leads them to home page of Indian Grid Certification Authority (IGCA) (<http://ca.garudaindia.in/>). Click on *Request Certificate* and acquire the required user/host certificate(s), *details are provided in IGCA section*.

b) Registered users are required to provide *User Id* and *Password* for logging into the Portal and access various facilities.

4.3 Job Management


User can submit their job, monitor the status and view output files using the Job Management interfaces. Types of job submission (Basic and Advanced) and Job information are covered under this section.

4.3.1. Basic Job Submission

This interface can be used to submit sequential as well as parallel jobs. The user should provide the following information:

1. Optional Job Name - User can provide a suitable (alias) name for their job.
2. Type of Job user want to execute,
3. Operating System – Required for their Job,
4. 'Have you reserved the Resources' - An optional parameter contains the Reservation Id's that can be used for job submission instead of choosing the Operating System/Processor parameter.
5. No. of processes required for the job - This parameter is only for the parallel applications that require more than one CPU.

6. Corresponding Executables – uploaded from either local or remote machine,
7. Input file, if required - The executable and the input file can either be uploaded from the local machine or can be selected from the Remote File List, if it is available in the Submit Node
8. STDIN - Required when the user wants to provide any inputs to the application during the runtime.
9. Optional Execution Time - Here the Execution Time is the anticipated job completion time.
10. Any Command Line arguments or Environment Variables, if required.
11. User Specific Output/ Error files - If the application generates output/error files other than standard output/error files and its entries should be separated by comma's or single empty space in case of multiple files.



The screenshot displays the 'Garuda - Job Submission' web interface. The header includes the CDAC logo and navigation links: Home, PSE, Paryavekshanam, and Logout. The main content area is divided into three tabs: Basic Submission (selected), Advanced Submission, and Job Info. On the left, there is a sidebar with links to Job Management, Resources, File Browser, Accounting, Myproxy, voms, Notices, Help, FAQs, and Partner site. The Basic Submission form contains the following fields and controls:

- Job Name: Text input field
- Job Type*: Dropdown menu with '--select--'
- Have you reserved the Resources: Checkbox (unchecked)
- Reservation Id: Text input field with 'No Reservation Ids' placeholder
- Operating System/ Processor*: Dropdown menu with '-----Select-----'
- Total number of processes: Dropdown menu with '1'
- Executable/Batch File*: Dropdown menu with 'Remote' and a text input field with '-- Select Remote File --'
- Input File: Dropdown menu with 'Remote' and a text input field with '-- Select Remote File --', plus an 'Add to List' button
- STDIN: Dropdown menu with 'Remote' and a text input field with '-- Select Remote File --'
- Execution Time: Four dropdown menus for '1 day', '0 hr', '0 min', and '0 sec', followed by '(Approx.)'
- Command line arguments: Text input field
- Environment Variables: Text input field
- User Specific Output File: Text input field

At the bottom of the form are 'Submit' and 'Reset' buttons.

Figure 5: Basic Job submission

All those fields marked with * are mandatory fields and should be filled before submitting a job. By clicking on *submit* button, the portal submits the job to GridWay Meta Scheduler, which then schedules the job for execution and returns the Job Id. The *Job Id* has to be noted for future reference to this job. In the event of unsuccessful submission, the corresponding error message is displayed.

4.3.2. Advanced Job Submission

This interface is provided for the user to submit their Sequential and Parallel Jobs. The difference from Basic job submission being: it is using GT4 Web Services components for submitting jobs to the Grid instead of Gridway as scheduler.



Garuda - Job Submission
India's National Grid Computing Initiative
Home | AGSG | PSE | Paryavekshanam | Logout

Basic Submission | **Advanced Submission** | Job Info

Job Management | Resources | File Browser | Accounting | Myproxy | voms | Notices | Help/User Manual | FAQs | Partner site

Click to Submit XML File

Select OS/Processor*

Select Executable*

Command Line Arguments

Input File

STDIN

Environment

Application Specific Output File

Application Specific Error File

Memory

No. Of Nodes*

No. Of Process*

[Help](#)

Note : All fields marked with * are mandatory

Figure 6: Advanced Submission Page

The user is provided with two modes in this interface:

1. Default mode - Portal creates the XML file for the user.
2. Second mode, recommended for advanced users - The user can provide their-own XML file as the executable, provided the required files are available in the submit node.

The user should provide the following information:

1. Operating System – Required for their Job,
2. Corresponding executables – uploaded from local/ remote machine.
3. Any Command Line arguments or Environment Variables, if required.
4. Input file(s), if required - The executable and the input file can either be uploaded from the local machine or can be selected from the Remote File List, if it is available in the Submit Node
5. STDIN - Required when the user wants to provide any inputs to the application during the runtime.
6. User Specific Output/Error files - If the application generates output/error files other than standard output/error files and its entries should be separated by comma's or single empty space in case of multiple files.
7. Total number of processes required for the job - This parameter is only for the parallel applications that require more than one CPU.
8. No. of Process - For parallel jobs, the total number of tasks or processes required to run the application and not in the hardware context of the CPUs available in the system.
9. No. of Nodes – For the user to achieve the optimal performance of the application and it allows the job scheduler to divide the number of tasks across the nodes.

The user can submit the job by clicking the *Submit* button, after the appropriate entries are done. The Grid Portal creates the XML file required for submitting job to Globus and returns *Job Id*. In the event of unsuccessful submission, the corresponding error message will appear at the bottom of this interface.

4.3.3. Job Info

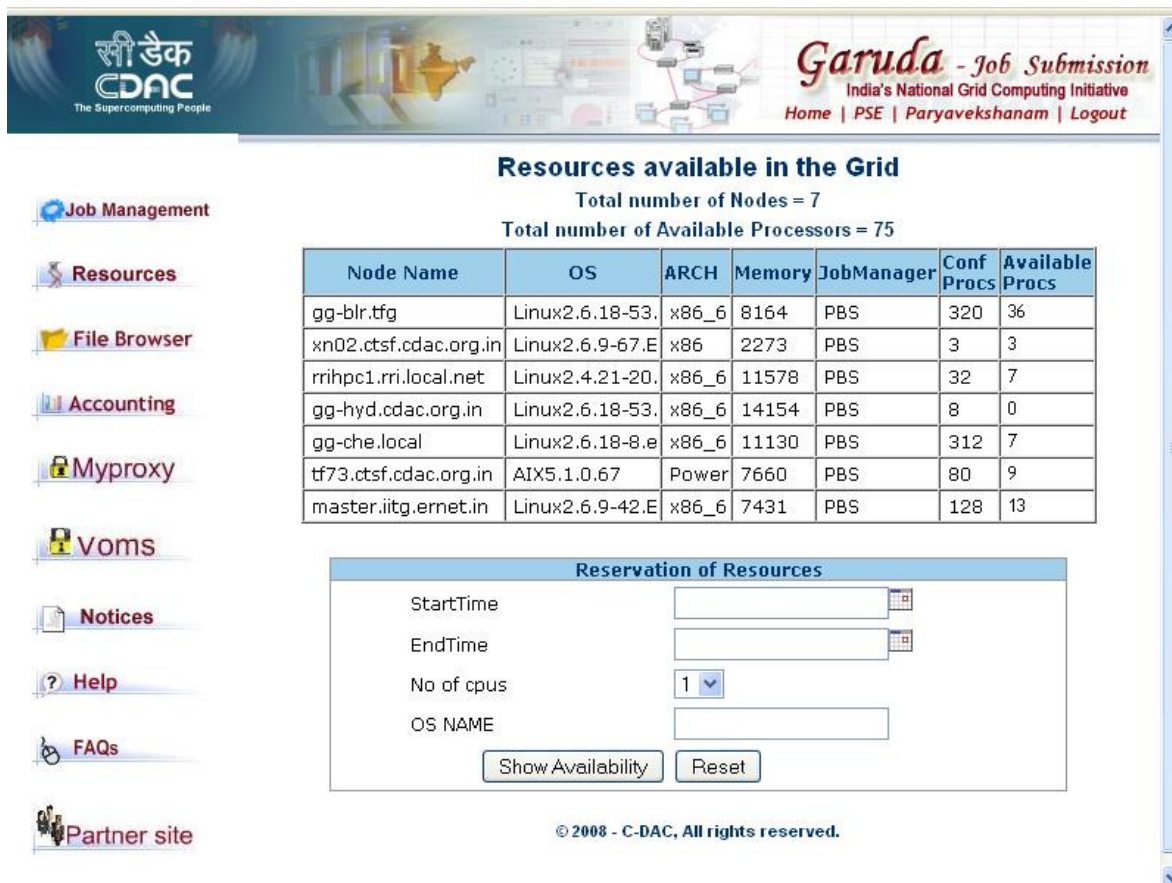
The user can view the status of the job submitted through Portal and the output file of the job by specifying the Job Id. The option for downloading the Output/ Error file is also provided, after the job execution. To cancel any of the queued jobs, the user has to select the job and click Cancel Job button, following which the acknowledgment for the job canceled is provided.



Figure 7: Job Information

4.4 Resources

The GridWay meta-scheduler provides the following information - Node Name, Head Node, OS, ARCH, Load Average, Status, Configured Process and Available Process. This information aids user to select a suitable cluster and reserve them in advance for job submission.



Resources available in the Grid

Total number of Nodes = 7
Total number of Available Processors = 75

| Node Name | OS | ARCH | Memory | JobManager | Conf Procs | Available Procs |
|-----------------------|-----------------|-------|--------|------------|------------|-----------------|
| gg-blr.tfg | Linux2.6.18-53. | x86_6 | 8164 | PBS | 320 | 36 |
| xn02.ctsf.cdac.org.in | Linux2.6.9-67.E | x86 | 2273 | PBS | 3 | 3 |
| rrihpc1.rii.local.net | Linux2.4.21-20. | x86_6 | 11578 | PBS | 32 | 7 |
| gg-hyd.cdac.org.in | Linux2.6.18-53. | x86_6 | 14154 | PBS | 8 | 0 |
| gg-che.local | Linux2.6.18-8.e | x86_6 | 11130 | PBS | 312 | 7 |
| tf73.ctsf.cdac.org.in | AIX5.1.0.67 | Power | 7660 | PBS | 80 | 9 |
| master.iitg.ernet.in | Linux2.6.9-42.E | x86_6 | 7431 | PBS | 128 | 13 |

Reservation of Resources

StartTime

EndTime

No of cpus

OS NAME

© 2008 - C-DAC, All rights reserved.

Figure 8: Resource Reservation

4.4.1 Reservation of Resources

The user can query the available free resources for reservation with respect to specified Period of Time, Number of Process and OS Name by clicking upon the 'Show Availability' button. Resource Name, OS Name, Conf Processes, Available Processes and Current Status of the resource are listed in tabular form.

Steps for reservation of resources:

1. Check the available free resources with valid parameters (Start Time and End Time – duration for which the resource needs to be reserved). The input fields No. of CPUs and OS entries are optional.

Example: starttime= 2009-04-02 17:06:53 endtime=2009-04-02 19:07:10 No. of CPUs=2 OS NAME=Linux

2. Choose the *Available Process* required for the job. Example: Available Procs = 4
3. Select the required resource from the available list of resources.
4. Book the resources for reserving a resource for the requested period of time and process.
5. The reserved resources can be modified/ canceled.
6. Once the reservation process is successfully completed, the Reservation Id is displayed and is made available in the Basic Job Submission page.

4.5 File Browser

For the logged-in user, the File Browser lists files, such as the uploaded executables and Input/Output/Error files, along with their size and last modified information. It also allows deletion of files.



The screenshot shows the Garuda Job Submission interface. The top banner includes the CDAC logo and navigation links: Home, PSE, Paryavekshanam, and Logout. The left sidebar contains links for Job Management, Resources, File Browser (highlighted), Accounting, Myproxy, Voms, Notices, Help, FAQs, and Partner site. The main content area is titled 'Remove Files' and displays a table of files.

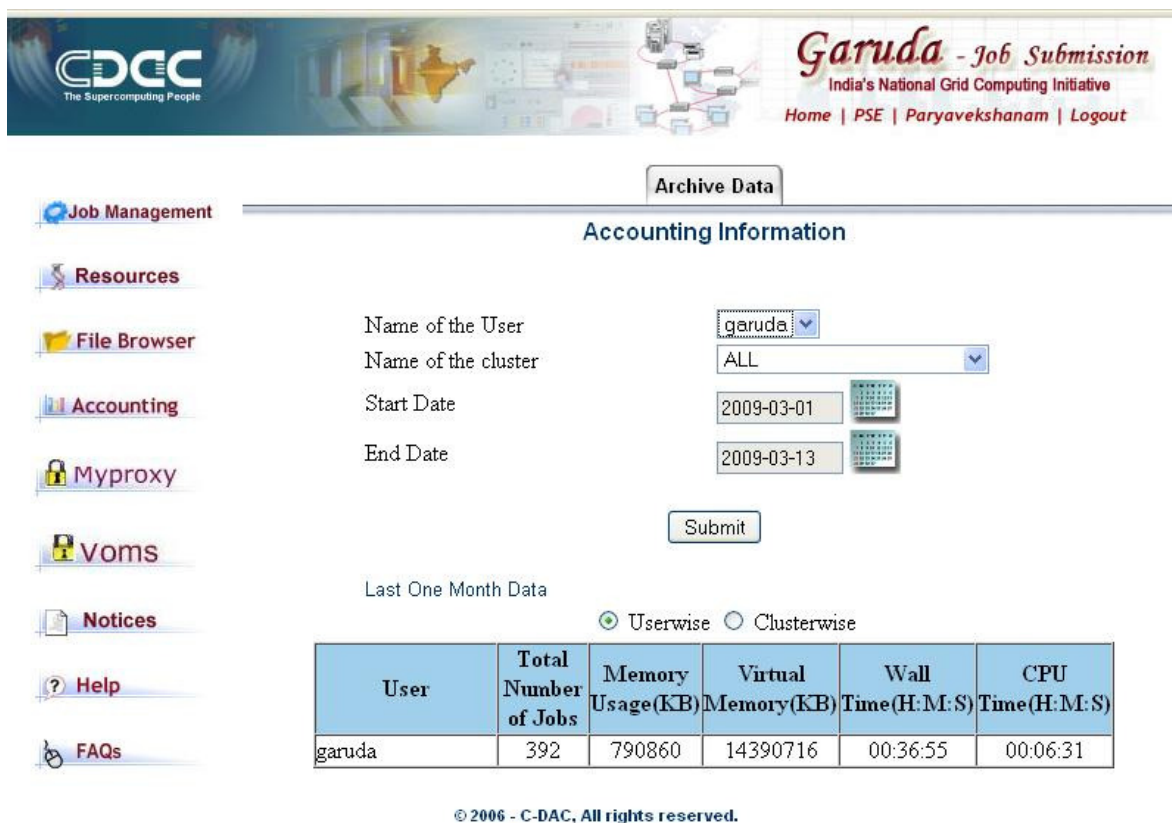
| <input type="checkbox"/> | File Name | Size in Bytes | Last Modified Date |
|--------------------------|------------------|---------------|--------------------------|
| <input type="checkbox"/> | BackUp2 | 12288 | Jun 26, 2007 9:59:29 AM |
| <input type="checkbox"/> | BIO | 4096 | Mar 3, 2009 9:47:54 AM |
| <input type="checkbox"/> | Execs | 4096 | Feb 24, 2009 5:31:29 PM |
| <input type="checkbox"/> | logs | 4096 | Jun 26, 2007 11:28:50 AM |
| <input type="checkbox"/> | RSLFilesDir | 4096 | Feb 23, 2009 5:32:23 PM |
| <input type="checkbox"/> | TEST | 4096 | Feb 25, 2009 5:22:02 PM |
| <input type="checkbox"/> | XML | 4096 | Mar 11, 2009 5:34:32 PM |
| <input type="checkbox"/> | 113048submit.xml | 1530 | Mar 12, 2009 2:42:47 PM |
| <input type="checkbox"/> | 113049submit.xml | 1487 | Mar 12, 2009 2:43:04 PM |
| <input type="checkbox"/> | 113050submit.xml | 1549 | Mar 12, 2009 2:45:54 PM |

Below the table are navigation buttons: << Prev, Delete, Reset, and Next >>. At the bottom, a small copyright notice reads: © 2006 - CDAC, All rights reserved.

Figure 9: File Browser

4.6 Accounting

This module provides Accounting information of the jobs that are submitted to GARUDA, such as no. of jobs submitted, and system parameters such as Memory usage, Virtual memory, Wall Time, and CPU time. Last one month data is displayed by default. The data can be viewed user-wise, or cluster-wise. The period also can be changed.



Archive Data

Accounting Information

Name of the User: garuda

Name of the cluster: ALL

Start Date: 2009-03-01

End Date: 2009-03-13

Submit

Last One Month Data

☒ Userwise ☐ Clusterwise

| User | Total Number of Jobs | Memory Usage(KB) | Virtual Memory(KB) | Wall Time(H:M:S) | CPU Time(H:M:S) |
|--------|----------------------|------------------|--------------------|------------------|-----------------|
| garuda | 392 | 790860 | 14390716 | 00:36:55 | 00:06:31 |

© 2006 - C-DAC, All rights reserved.

Figure 10: Accounting

4.7 MyProxy

MyProxy allows user to upload their Globus Certificates into Myproxy Server and the same can be used for initializing the Grid proxy on the Grid. The advantage of using MyProxy is that the user need not keep their Certificates on Grid; instead they can upload and get certificate from the secured Myproxy Server whenever required.

The user requires two files namely, `usercert.pem` and `userkey.pem`, on their local machine to use Myproxy Init. Users can submit jobs to the Grid only after initializing the proxy.

If the certificate has been already generated for you, but you do not have access to the above-mentioned files, you can download it from GridFS machine (from `$HOME/.globus` directory) using `winscp/scp`.



The screenshot shows the Garuda - Job Submission web interface. The header includes the CDAC logo and the text "Garuda - Job Submission India's National Grid Computing Initiative". Below the header, there are tabs for "MyProxy Init" and "MyProxy Get". The "MyProxy Init" tab is active, displaying a form titled "Please fill login information". The form contains the following fields:

- User Name:
- GridProxyPassPhrase:
- MyProxy PassPhrase:
- Re enter MyProxy PassPhrase:
- User Cert File:
- User Key File:
- Proxy Life Time: Hrs

At the bottom of the form are two buttons: "Upload Proxy" and "Reset". A note at the bottom of the page states: "Note : All fields are mandatory".

Figure 11: MyProxy

4.7.1 MyProxy Init

By default, the "Myproxy Init" option is enabled for the user. Upload proxy by entering valid inputs - User name, Grid-proxy Passphrase, User certificate file (`usercert.pem`), User key file (`userkey.pem`) and Proxy life time (168 hours is the default value).

4.7.2 MyProxy Get

Grid proxy will be initialized on the Grid head node by providing the inputs - User name, Myproxy Passphrase and Life time of the certificate.

4.8 VOMS Proxy

The Virtual Organization Management System (VOMS) allows users to belong to Virtual Organizations (VOs), thereby allowing them to utilize resources earmarked for those VOs. The user can initialize the VOMS Proxy on Grid head node for the valid VO name and its Roles. User is required to input VO name, Role, and Grid passphrase, and Upload the Globus Certificates, vomses file. *vomses* file is nothing but a non-standard location of configuration file. This file consists of VO name and Host Key information.



VOMS PROXY

Voms Information

submit voms proxy Information

| | | |
|-----------------|---|--|
| VO Name | <input type="text" value="application"/> | Request for vo? |
| Role | <input type="button" value="No Specific Role"/> <input type="button" value="programmer"/> | |
| Vomses | <input type="text" value="C:\Documents and"/> | <input type="button" value="Browse..."/> |
| Grid PassPhrase | <input type="password" value="....."/> | |
| User Cert File | <input type="text" value="C:\Documents and"/> | <input type="button" value="Browse..."/> |
| User Key File | <input type="text" value="C:\Documents and"/> | <input type="button" value="Browse..."/> |
| Time Limit | <input type="text" value="168"/> | <input type="text" value="Hrs"/> |

Note : All fields are mandatory

© 2006 - C-DAC, All rights reserved.

Figure 12: VOMS Proxy

The user can also request for a new VO by using "Request for VO" link. VOMS proxy initialization with Multiple roles is provided to the user, by selecting more than one entry on the Role combo box.

"No specific Role" entry in the Role combo box is mainly for the VO which is not having any role or no role assigned by the VOMS admin. The default Time Limit of 168 hours can be modified by overwriting on this field. The current status of the VOMS proxy information is available to the user by clicking on Proxy Info button.

4.9 Partners Site

This interface provides the facility for GARUDA Partners to create and update their web pages. Currently, this facility is enabled for only a few Partners. The remaining Partners can activate this facility by sending a request mail to grid-help@cdacb.ernet.in.

4.10 Notices

This interface provides information about current updates available in the GARUDA Grid. If there are issues with any cluster, it is mentioned in Notices.

5. Data Management

To enable data oriented applications, GARUDA provides an integrated and distributed data storage architecture by deploying the Storage Resource Broker (SRB) from Nirvana. SRB creates and maintains a Global Namespace across multiple heterogeneous and distributed storage systems in the Grid so that users can access their data from any where at any time.

5.1 Data Grid Solution

GARUDA Storage Resource Manager (GSRM) is a peer to peer data grid solution for SOA based GARUDA. An SRM can be defined as a middle ware component that manages the dynamic use of storage resources on the Grid. This means that space can be allocated dynamically to a client, and that the decision of which files to keep in the storage space is controlled dynamically by the SRM. GSRM is a disk based SRM implementation, following SRM version 2.2 specifications. It adheres to Open Grid Forum standards. It is based on open source Disk Pool Manager (DPM) version 1.7.2.

The GSRM version1.0 supports the following features:

1. Provides single point access to distributed and heterogeneous storage resources for users by providing *Global Namespace*.
2. Provides *File and Directory Management* - Upload /download of files in spaces, pinning & release of files.
3. Supports GSI, Kerberos 5, Access Control Lists, and VOMS *Security Mechanism* and supports secure data transfers.
4. GSRM provides dynamic *Space Management* reserving & release of storage spaces and setting lifetime for the files.

5. Provides *User friendly interfaces* to access its services.
6. Provides *Interoperability* with other SRM implementations, based on SRMv2.2 specifications like Best Man – Berkeley Storage Manager, Castor-SRM, dCache-SRM, StoRM - Storage Resource Manager. Users can access GSRM storage services using any of the above mentioned client interfaces.

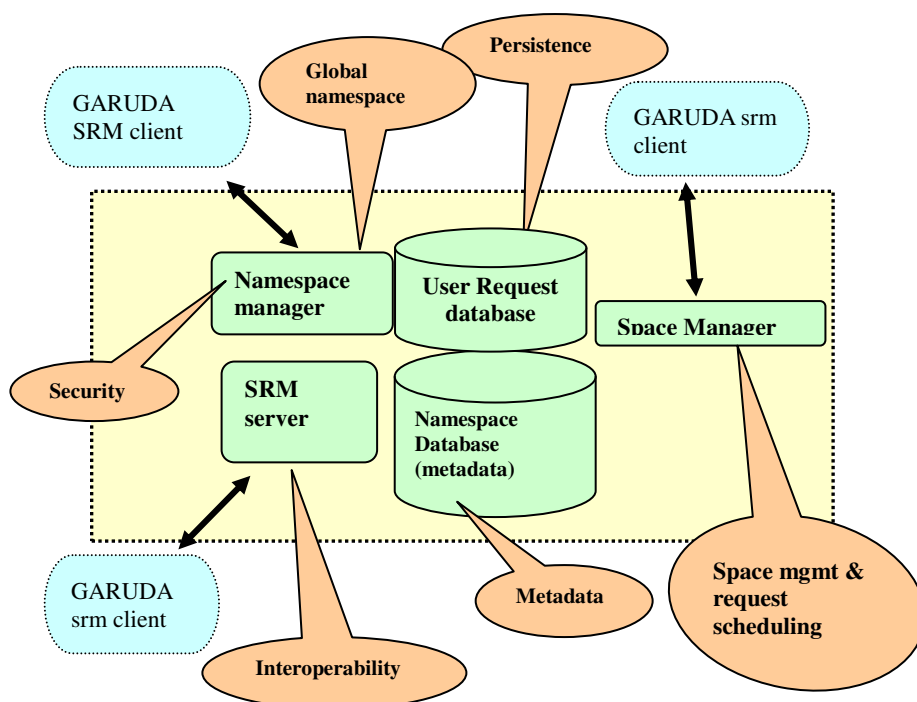


Figure 13: GSRM Components

Figure shows the various GSRM components and their feature mapping. GSRM components are depicted with green boxes and their corresponding functionality with orange boxes.

5.2 GSRM Accessibility in GARUDA:

GARUDA services are available to users at GARUDA head node (currently only at Bangalore—gridfs (192.168.60.40)). The storage systems supported in this version of GSRM(v1.0) are disk based storage. Users can either access GSRM storage through command

line clients installed at gridfs . So, users can download the input files from GSRM before submitting the job and can also store the generated output files in it after the completion of the job. GSRM will also be accessible through GARUDA portal in future. Figure 2 depicts the GSRM integration and interaction with other GARUDA middleware components with respect to the job submission by the user.

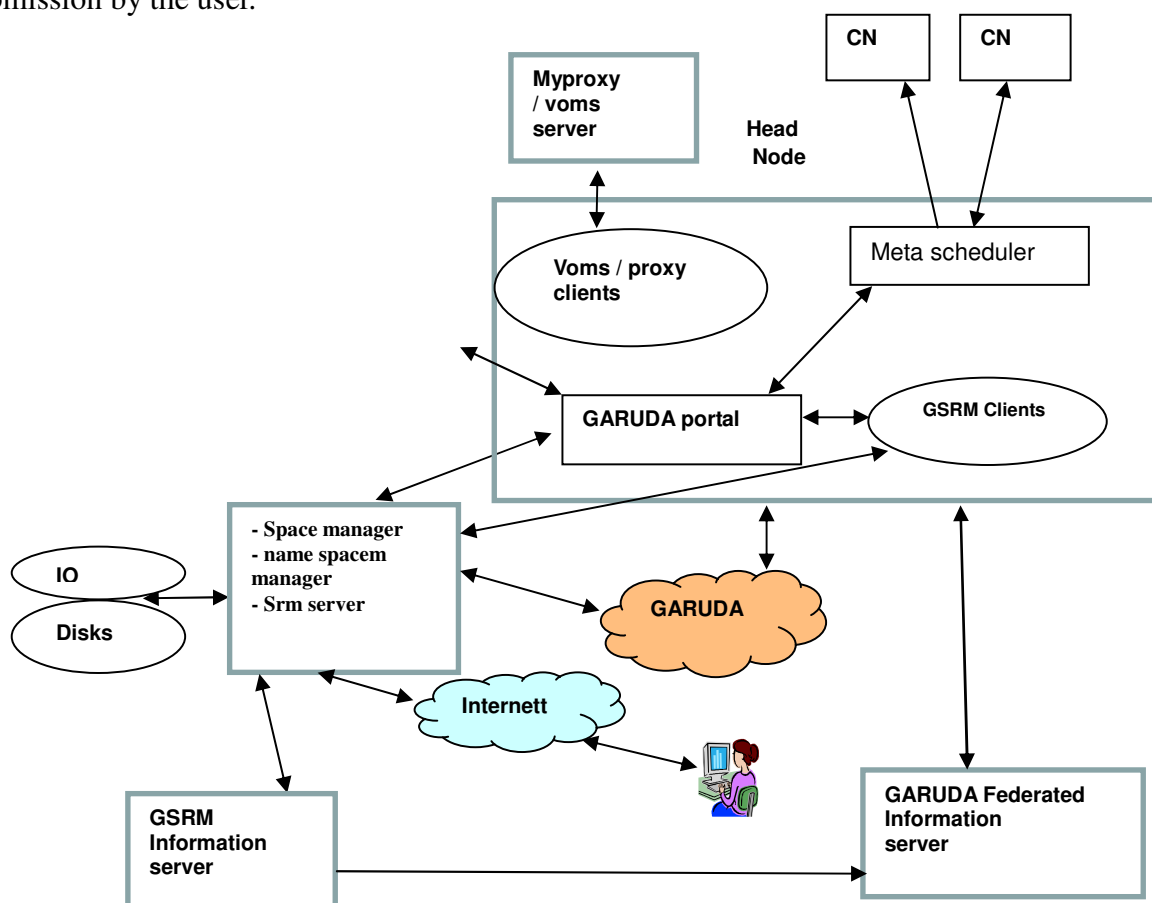


Figure 14: GSRM Usage by GARUDA Users

Steps to be followed to access GSRM from gridfs:

- Login to gridfs(192.168.60.40)
- Upload your IGCA user certificates
- Initialize proxy with grid-proxy-init

- Set environmental variables, respectively for whichever client to be used.
- Run the SRM commands

GSRM services can be accessed by the following client interfaces:

- **Command Line** :The various command line interfaces available to access GSRM services are StoRM (clientSRM commands), BestMan and dpm command line interfaces.
- **APIs** : GSRM supports DPM 'C' APIs for C based applications and BestMan 'Java' APIs for Java based applications..
- **Web Interface**: DPM web interface is available to access GSRM services through browser.

GSRM Access points

Currently, GSRM services can be accessed logging into GARUDA Bangalore head node gridfs (192.168.60.40) or pvfs2 (172.20.1.81). SRM client installations are available on both the above mentioned nodes.

- **pvfs2(172.20.1.81)** node should be used to just test all the available SRM client interfaces like StoRM,DPm,BestMan.
- **gridfs (192.168.60.40)** node should, if the user wishes to use GSRM storage for job execution. Users can download /Upload , input/output files into GSRM while submitting jobs from gridfs.

5.3 Pre-requisites for using GSRM services

- A valid user IGCA certificate (Valid user certificate can be obtained through IGCA web site :: <https://ca.garudaindia.in/>)
- A SRM endpoint to contact (<http://xn05.ctsf.cdac.org.in:8446>)
- Unix Login for GARUDA Grid Head node gridfs (mail to : grid-help@cdacb.ernet.in) or pvfs2 (mail to : rt-gds@cdacb.ernet.in)
- Authorization to access SRM storage resources.(mail to : rt-gds@cdacb.ernet.in)
- SRM client interface Installation (Any of the above mentioned SRM client installation)

Following **Access mechanisms** are available at above mentioned nodes to access GSRM:

1. **gridfs(192.168.60.40)** : gridfs is the Bangalore GARUDA head node. GSRM services can be accessed from here using StoRM command line interface.

If the user wants to use the clientSRM (StoRM Clients) from gridfs machine

- Create a valid user proxy using grid-proxy-init
- Set the env variable for Globus location path

```
export GLOBUS_LOCATION= GLOBUS_LOCATION:/usr/local/GARUDA/GLOBUS-4.0.7/
export PATH=$PATH:/opt/gsrn-client/srmv2storm/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/gsrn-client/cgsi_soap/lib
```

- Run the clientSRM command

2. **pvfs2 (172.20.1.81)**: pvfs2 is the GSRM testing node with the following client interfaces installed.

- StoRM, Bestman and dpm command line interfaces

- Bestman Java APIs
- DPM C APIs

3. GSRM Web Client is accessible from any of the user machines reachable to GSRM server (xn05.ctsf.cdac.org.in), using URL -- <https://xn05.ctsf.cdac.org.in/>

5.4 GSRM Client Interfaces

GSRM is inter operable with other SRM implementations, its services can be accessed by clients of any other SRM implementation. Following are the client installations that can be made available to users with the supported user interfaces to access GSRM services are as follows:

| Sl. | SRM Clients | User Interfaces Installations |
|-----|-------------|--|
| 1 | StoRM | <ul style="list-style-type: none"> • Command Line |
| 2 | DPM | <ul style="list-style-type: none"> • Command Line • C APIs • Web Client |
| 3 | BestMan | <ul style="list-style-type: none"> • Command Line • Java APIs |

Given below are the few sample srm client commands:

StoRM Command Line Client

1. StoRM command line client format:

clientSRM <requestName> <requestOptions>

2. To get help for clientSRM commands:

clientSRM -h

3. Command to ping to GSRM server:

clientSRM ping -e <GSRM end point>

DPM C API

Table 1 shows the list of DPM ‘C’ APIs currently supported by GSRM

| <u>COMMAND</u> | <u>DESCRIPTION</u> |
|-----------------------|--|
| dpm_copy | copy a set of existing files |
| dpm_get | make a set of existing files available for I/O |
| dpm_getstatus_copyreq | get status for a dpm_copy request |
| dpm_getstatus_getreq | get status for a dpm_get request |
| dpm_getstatus_putreq | get status for a dpm_put request |
| dpm_put | make a set of existing files available for I/O |
| dpm_putdone | mark a set of files as complete |

Table 1: List of DPM ‘C’ APIs

Bestman Command Line Clients

1. Command to ping to GSRM server

```
srm-ping –serviceurl httpg://xn05.ctsf.cdac.org.in:8446/dpm/ctsf.cdac.org.in/home/garuda
```

2. Upload file to GSRM server

```
srm-copy <src url> <target url> <service url>
```

NOTE : For details on the GSRM client interfaces click on http://180.149.48.139/garuda-wiki/index.php/Main_Page

6. PARYAVEKSHANAM

Version 3.0

Paryavekshanam monitors all major components of the grid – namely computing nodes, network, globus components, jobs, storage and software. It supports discovery, monitoring, logging and notification for all the grid resources and services. Users can have detailed information about the machine's network stability, bandwidth utilization, memory utilization, node availability, Globus strength, CPU Load and also the Link status.

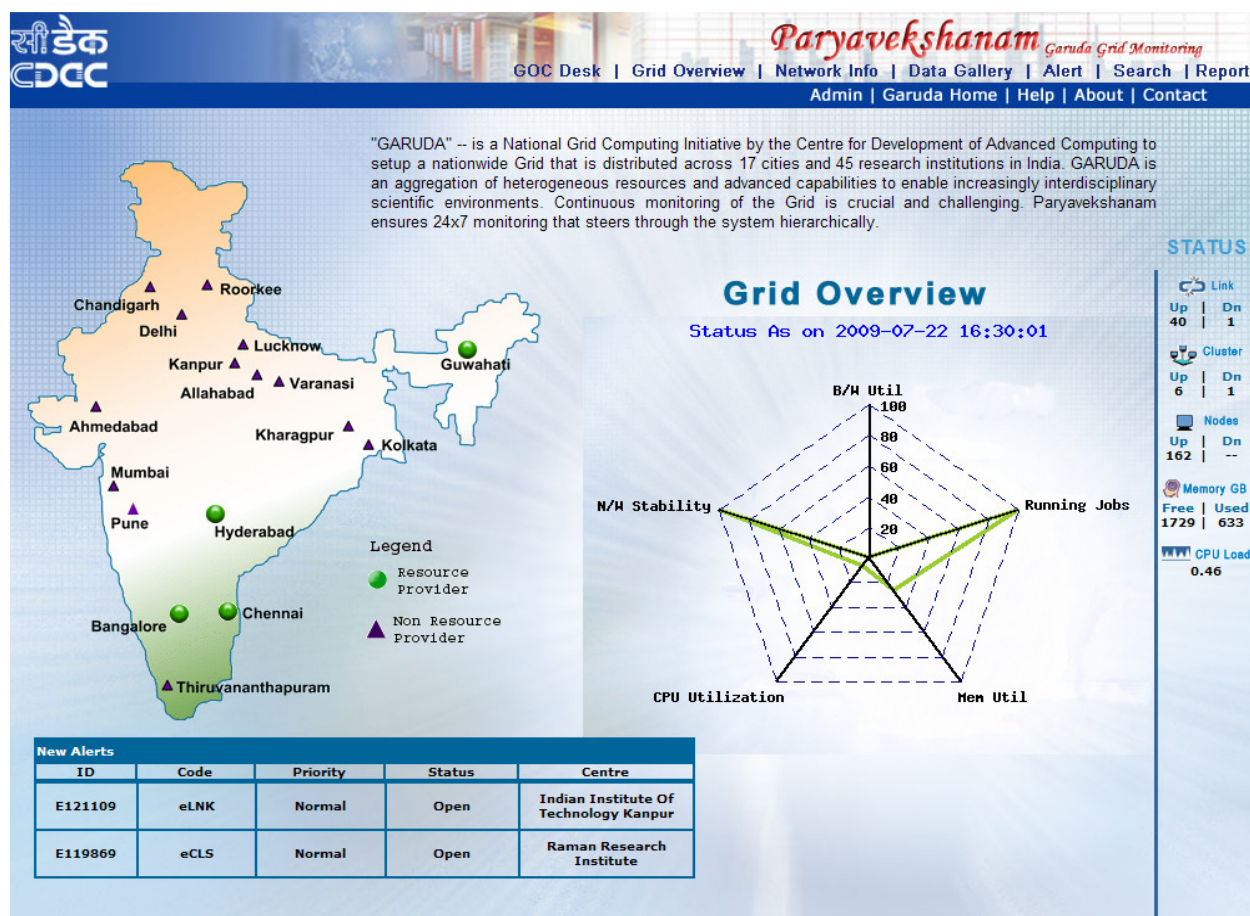


Figure 15: Home page of Paryavekshanam

A first level fault analysis can be carried out and the concerned local centre will be informed / alerted to rectify the problem immediately. (For problem reporting, please refer the next section). The Home page indicates operational status of the grid at individual cities. The users can drill down to view the status of individual resource at site level.

6.1 Pre-requisites for using Paryavekshanam

Supported browser versions

- Internet Explorer 6.0 and above,
- Mozilla Firefox 2.0 and above

6.2 Accessing Paryavekshanam

Type <http://192.168.60.70/gridmon/GRID/gridmon.php> in the address bar of the web browser to invoke Paryavekshanam.

6.3 Features

1. *Rich user interface* providing different outlooks - hierarchical drill down, instant status, quick jump links or detailed graphs.
2. *Escalation* of service failures / degradations through alert messages
3. *Archival* of data for report generation and analysis
4. *Search* facility

6.4 Web Pages

1. *Paryavekshanam* - Paryavekshanam home page.
2. *GOC Desk* - Graphical representation of Grid parameters.

3. *Grid Overview* – Tabular information about the monitored parameters of the Grid.
4. *Network Info* – Provides centre-wise network information of the Grid.
5. *Data Gallery* – Graphical representation of archived data.
6. *Alert page* – Detailed alert message listing.
7. *Search page* – Facility for searching of resources on the Grid.
8. *Report page* – Reports of job accounting and resources.
9. *Admin page* – Facility for adding new resource provider sites.
10. *About Us* - Information regarding GARUDA grid monitoring system.
11. *Contact Us* - Contact information for reporting the bugs/queries related to Paryavekshanam.


| <div>  <div> Paryavekshanam <small>Garuda Grid Monitoring</small> </div> <div> GOC Desk Grid Overview Network Info Data Gallery Alert Search Report </div> <div> Admin Garuda Home Help About Contact </div> </div> | | | | | | | | |
|--|--------------------------|-------------|------------------------|-------------------|-----------------------|-------|----------|----------|
| Grid Overview | | | | | | | | |
| Status as on 2009-07-21 12:40:02 | | | | | | | | |
| Cluster Names | Centres | Nodes Avail | CPU Util | Total Memory [GB] | Available Memory [GB] | State | Run Jobs | Que Jobs |
| Hyderabad PARAM Cluster | C-DAC_Hyderabad | 8 | <div><div></div></div> | 23.77 | 15.34 | up | 0 | 0 |
| NTF cluster | C-DAC_Knowledge_Park | 50 | <div><div></div></div> | 447.78 | 393.02 | up | 0 | 0 |
| Rrihpc1 | Raman_Research_Institute | 0 | <div><div></div></div> | | | down | | |
| orang | IIT_Guwahati | 15 | <div><div></div></div> | 424.49 | 412.10 | up | 12 | 0 |
| CDAC Garuda Cluster | C-DAC_Chennai | 42 | <div><div></div></div> | 689.86 | 624.15 | up | 0 | 0 |
| CDAC Hyderabad Cluster | C-DAC_Hyderabad | 10 | <div><div></div></div> | 188.26 | 79.81 | up | 0 | 0 |
| Garuda Grid Bangalore | C-DAC_Knowledge_Park | 37 | <div><div></div></div> | 587.83 | 200.98 | up | 0 | 0 |
| Note: The data provided here is collected from centralized Information Server | | | | | | | | |

Figure 16: Grid Overview Page


























|   | | | | | |
|---|---|------------------|-----------------|--------------------------|----------------------|
| GOC Desk Grid Overview Network Info Data Gallery Alert Search Admin Garuda Home Help Aboutus Contactus | | | | | |
| Network Overview | | | | | |
| Status as on 2007-07-30 19:00:01 | | | | | |
| City/Centres | Link Status | B/w Avail (mbps) | B/w Used (kbps) | Packet Loss (percentage) | Round Trip Time (ms) |
| Ahmedabad | | | | | |
| Physical Research Laboratory |  | 10.0 | 1.1 | 0.0 | 36.3 |
| Space Application Centre |  | 100.0 | 0.0 | 0.0 | 0.0 |
| Institute for Plasma Research |  | 100.0 | 0.7 | 0.0 | 36.7 |
| Allahabad | | | | | |
| Harish-Chandra Research Institute |  | 10.0 | 0.6 | 0.0 | 51.5 |
| Motilal Nehru National Institute Of Technology |  | 10.0 | 0.7 | 0.0 | 52.6 |
| Bangalore | | | | | |
| C-DAC Knowledge Park |  | 100.0 | 249.9 | 0.0 | -- |
| C-DAC Electronic City |  | 10.0 | 1.3 | 0.0 | 3.7 |
| Indian Institute Of Science |  | 100.0 | 1.4 | 0.0 | 3.4 |
| Indian Institute Of Astrophysics |  | 10.0 | 5.3 | 0.0 | 3.6 |
| Raman Research Institute |  | 100.0 | 7.8 | 0.0 | 3.5 |
| National Center for Biological Science |  | 10.0 | 0.0 | 0.0 | 7.4 |
| C-DAC Knowledge Park L2 Link |  | 100.0 | 79.2 | 0.0 | 0.6 |
| Chandigarh | | | | | |
| Institute Of Microbial Technology |  | 10.0 | 0.6 | 0.0 | 48.3 |
| Punjab Engineering College |  | 100.0 | 0.8 | 0.0 | 36.1 |
| Chennai | | | | | |
| Madras Inst. Of Tech. |  | 100.0 | 1.1 | 0.0 | 7.8 |
| C-DAC Chennai |  | 10.0 | 0.0 | 0.0 | 7.0 |
| Institute of Mathematical Sciences |  | 10.0 | 0.6 | 0.0 | 7.8 |
| IIT Chennai |  | 10.0 | 171.5 | 0.0 | 6.6 |
| Delhi | | | | | |
| ERNET |  | 10.0 | 8.1 | 0.0 | 33.5 |
| IGIB |  | 100.0 | 0.7 | 0.0 | 33.2 |
| JNU |  | 100.0 | 0.0 | 0.0 | 0.0 |
| IIT Delhi |  | 100.0 | 50.4 | 0.0 | 33.8 |

Figure 17: Network Overview Page

Here, green dots represent available links, and gray dots represent links unreachable.



Paryavekshanam
Garuda Grid Monitoring

[GOC Desk](#) | [Grid Overview](#) | [Network Info](#) | [Data Gallery](#) | [Alert](#) | [Search](#)
[Admin](#) | [Garuda Home](#) | [Help](#) | [Aboutus](#) | [Contactus](#)

View Alerts

Error ID :

Status : -- v

Centre : -- v

Date -- v (yyyy-mm-dd)
 and (yyyy-mm-dd)

Show 20 v records per page

Showing Page 1 of 4 | [First](#) | [Previous](#) | [Next](#) | [Last](#) |

| Err ID | Raised on Date | Resource Effected | Centre | Description | Status |
|---------|---------------------|--------------------------------|-----------------------|-----------------|--------|
| E102303 | 2007-07-30 18:21:02 | mitcluster.mitgrid.chn.cdac.in | Madras_Inst._Of_Tech. | Cluster is Down | open |
| E102300 | 2007-07-30 16:40:45 | mitnode46.mitgrid.chn.cdac.in | Madras_Inst._Of_Tech. | Node is Down | open |
| E102297 | 2007-07-30 12:41:09 | hyd01.hardware.cdac.ernet.in | C-DAC_Hyderabad | Cluster is Down | open |
| E102294 | 2007-07-30 11:31:04 | mitnode32.mitgrid.chn.cdac.in | Madras_Inst._Of_Tech. | Node is Down | open |
| E102296 | 2007-07-30 11:31:04 | mitnode55.mitgrid.chn.cdac.in | Madras_Inst._Of_Tech. | Node is Down | open |

Figure 18: Alert Page

Alert helps in finding and tracking all the error messages generated by the system. The alerts are listed in a table along with details - status of the alert, Error Id, date on which alert was raised etc.

7. COMPILER SERVICE

The SOA compiler is aimed for building applications for Grid. The tool provides a simple GUI to build applications easily, thereby hiding the complexities involved to access the remote grid resources and avoid hassles of command line interactions.

The tool can be divided into three parts, namely: client GUI, Compiler Server and Compiler Web Service. The Client GUI runs on the user desktop. When Compiler service is accessed by the user, the required client GUI Java classes are downloaded on their machine and contacts the Compiler Server for subsequent operations. The inputs from user for building their application is stored on the Compiler Server, which in turn invokes the Compiler Web Service for the compilation on the user-specified Grid resource.

7.1 Pre-requisites for using SOA Compiler

1. Java Run Time Environment (JDK1.6+)
2. Web Browser with Java web start support

7.2 Accessing SOA Compiler

To access the tool, the user requires the Java Run Time Environment enabled standard web browser. When provided with complete URL (<http://192.168.60.116:9090/compiler>) of Compiler server, Java Web start downloads the entire Compiler GUI Java classes on the user's desktop machine. Java Run Time Environment in turn builds the Compiler GUI.

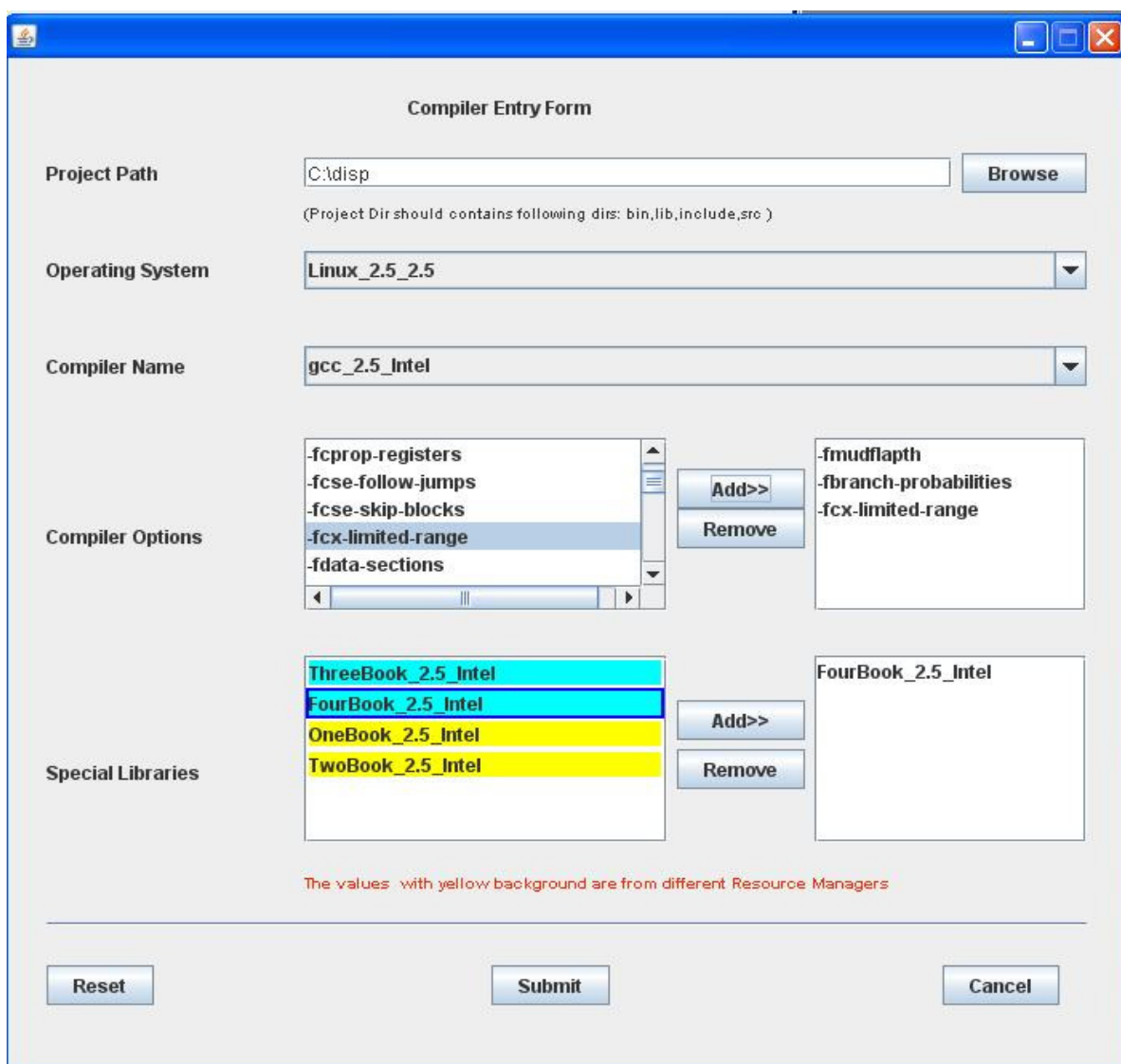
7.3 Compiler GUI

When the user invokes the tool, the following inputs have to be provided:

- Application Project Directory
- Operating System - User can select the platform to build their source application. The list of compilers along with their options and special libraries available on that

Operating System is populated.

- **Compiler Name** - User can choose from the list of compilers available.
- **Compiler Options** - Based on the compiler name selected, the user can select from multiple compiler options populated on the left side of the list box.
- **Special Libraries** - User can select particular special library required by the application.



The screenshot shows the 'Compiler Entry Form' window. It contains the following fields and controls:

- Project Path:** A text box with 'C:\disp' and a 'Browse' button. Below it, a note says '(Project Dir should contains following dirs: bin,lib,include,src)'.
- Operating System:** A dropdown menu showing 'Linux_2.5_2.5'.
- Compiler Name:** A dropdown menu showing 'gcc_2.5_Intel'.
- Compiler Options:** A list box on the left containing: '-fcprop-registers', '-fcse-follow-jumps', '-fcse-skip-blocks', '-fcx-limited-range' (highlighted), and '-fddata-sections'. To its right are 'Add>>' and 'Remove' buttons. A second list box on the right contains: '-fmudflapth', '-fbranch-probabilities', and '-fcx-limited-range'.
- Special Libraries:** A list box containing: 'ThreeBook_2.5_Intel', 'FourBook_2.5_Intel' (highlighted), 'OneBook_2.5_Intel' (highlighted), and 'TwoBook_2.5_Intel' (highlighted). To its right are 'Add>>' and 'Remove' buttons. A second list box on the right contains 'FourBook_2.5_Intel'.

At the bottom, there is a red text note: 'The values with yellow background are from different Resource Managers'. At the very bottom are 'Reset', 'Submit', and 'Cancel' buttons.

Figure 19: Compiler GUI

The users are required to adhere to following directory structure. Application Parent Dir- src/
,bin/, lib/ ,include/

7.4 Makefile Generation

The options selected by the user are validated, the inbuilt Makefile Generator creates the makefile. A simple text editor displays the makefile for user to edit and save. The errors, if any while generating the makefile, are displayed to the user.

The application project directory along with its makefile is zipped and transferred using Java Secured Shell API's (J2ssh) to the Compiler Server for further operations.

7.5 Building the application

The Compiler Server unzips the application source and transfers it to the selected Grid resource*. The Compiler Web Service is invoked with makefile as its input. It is submitted as a Grid job and generates the executable. Compiler Web Service returns either the executables back to the Compiler Server or the error messages generated. The user will be informed about the status of compilation.

* The Grid resource is decided based on the user inputs – OS, Compiler name, Compiler option and Special library selected.

7.6 Accessing Compiler Web Service

The SOA Compiler Web service is created for compiling user applications on user specified GARUDA cluster. The SOA Compiler Tool in turn uses Compiler web service for compilation.

7.6.1 Pre-requisites for using Compiler Web Service

- Globus-4.0.7+
- Valid proxy certificate from *Login service*.

Inputs

- *Target Machine name* Machine on which compilation has to be done
- *Zip Filename* Application filename in zipped format
- *Zip File Location* Location of zipped Application file name

Output

- *Compilation Status* Whether compilation was successful or not

7.6.2 Accessing Compiler Web Service

Compiler service can be found at the following URL which can be accessed from the respective clients:

<https://gg-blr.tfg:8443/wsrf/services/garuda/CompilerService>

8. UTILITY SERVICES

8.1 Login Web Service

The SOA Login Web service is created with purpose of unifying the login module in all the tools used. Application or tools requiring to verify the user's identity can use this web service in *their code*.

8.1.1 Pre-requisites for using Login Web Service

- Globus-4.0.7+
- MyProxy Server should be configured to automatically extend VO attributes from VO server

It exposes 3 methods for providing various accounting information.

1) login

This method is for logging in to the GARUDA.

Inputs

- | | |
|--------------------|---|
| • <i>user name</i> | MyProxy User Name |
| • <i>password</i> | MyProxy Password |
| • <i>life time</i> | Indicates how long is the proxy's life time |

Output

- | | |
|----------------------|---|
| • Proxy string | Proxy issued by the My proxy server |
| • Login status | Indicates the status of the operation |
| • Last Login Time | Gives when this user was last logged in |
| • Current Login Time | Gives users t logging in time |

2) uploadProxy

This method uploads a proxy that is generated using other tools, to the MyProxy Server.

Inputs

- *user name* MyProxy User Name
- *password* MyProxy Password
- *proxyBytes* Existing proxy file is given as byte array

Output

- *uploadStatus* Indicates the status of the operation

3) storeCredential

This method is used for uploading the credentials that is the PKCS12 certificate directly to the MyProxy Server. It will convert the PKCS12 to certificate and stores in server for users to download the proxy until it expires.

Inputs

- *user name* MyProxy User Name
- *password* MyProxy Password
- *p12Bytes* PKCS12 file as byte array

Output

- *storeStatus* Indicates the status of the operation

8.1.2 Accessing Login Web Service

Login service can be found at the following URL which can be accessed from the respective clients:

<https://gg-blr.tfg:8443/wsrf/services/garuda/LoginService>

User wants to access the service can either write a client or they can use the API implementation of the client supporting most common options. The details of the API is given as Java docs.

8.2 Accounting Web Service

The SOA Accounting Web service is created for giving accounting information to other applications like GARUDA Access Portal, Prayaveekshanam, etc. It exposes functions for

8.2.1 Pre-requisites for using Accounting Web Service

- Globus-4.0.7+
- Gridway – Customization should be done with gridway job submission module.
- Updater service needs to be deployed on all the cluster head nodes.

Services: It exposes 5 methods for providing various accounting information.

1. **getExecutionInfo**

Input: Job ID

Output: Returns object containing the information about the job execution.

2. **getExecutionHistory**

Inputs: A filter object containing - Start date of job execution, End date of job execution, Cluster on which the job was executed, DN

Output: Returns an array of objects containing the information about the job execution.

3. **getJobInfo**

Input: Job ID

Output: Returns object containing the information about that Job.

4. **getJobHistory**

Input: A filter object containing - Start date of job execution, End date of job execution,

Cluster on which the job was executed, DN

Output: Returns an array of objects containing the information about the history of jobs for the given duration.

5. getTotalResourceUsage

Input: A filter object containing - Start date of job execution, End date of job execution, Cluster on which the job was executed, DN

Output: Returns object containing the aggregate usage of resources according to the filter set.

8.2.2 Database Interactions:

Updater Service installed on each head node is called by Gridway after each job is completed which will update database server Accounting service (which can be installed on multiple locations) will access the database for getting the information.

8.2.3 Accessing Accounting Web Service

Accounting service can be found at the following URL which can access with the respective clients:

<https://gg-blr.tfg:8443/wsrf/services/garuda/AccountingService>

9. GARUDA Request Tracking-RT

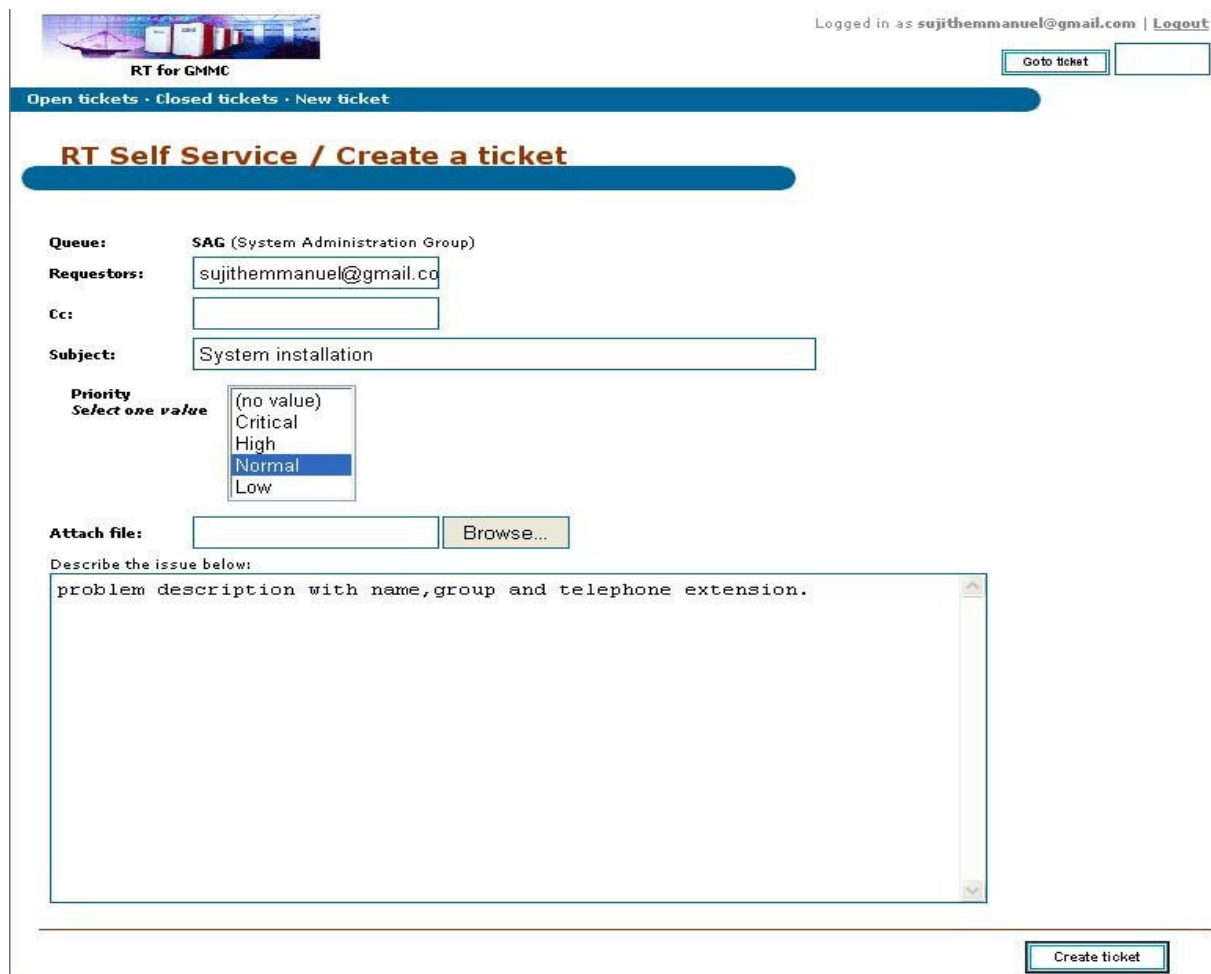
If a user encounters difficulty while job submitting, it can be reported using GARUDA Request Tracker. Initial analysis of the problem can be done using Paryavekshanam (Refer previous Section) or the user can directly report the issue to RT (Request Tracker).

The GARUDA RT manages the identification, prioritization, assignment, resolution and notification required for the resolution of issues in the GARUDA Grid. It keeps track of each ticket's full history and meta data, including time spent per action, due dates, and estimated time to completion.

Users can submit the issues or defects just by sending email or through the web interface to GARUDA RT. The GARUDA RT supports problem reporting for various functional categories of the GARUDA Grid including Job Portal, Grid Deployment and Grid IDE. The system will take care of thanking the issue reporters for their message, automatically routing it to the appropriate staff and making sure all future correspondence gets to the right place.

For raising a ticket, one has to be a valid user of GARUDA RT. To get registered, the user has to send a request to support@cdacb.ernet.in or support@garudaindia.in.

After the user procures ID and password, the user can login at <https://gridsupport.garudaindia.in> and raise a ticket against different queues available as shown below in Figure.



The screenshot shows the 'RT Self Service / Create a ticket' interface. At the top, it says 'RT for GMMC' and 'Logged in as sujithemmanuel@gmail.com | Logout'. There are links for 'Open tickets', 'Closed tickets', and 'New ticket'. The form fields are as follows:

- Queue:** SAG (System Administration Group)
- Requestors:** sujithemmanuel@gmail.co
- Cc:** (empty field)
- Subject:** System installation
- Priority:** A dropdown menu is open showing options: (no value), Critical, High, Normal (selected), and Low.
- Attach file:** (empty field) with a 'Browse...' button.
- Describe the issue below:** A text area containing the text 'problem description with name,group and telephone extension.'

A 'Create ticket' button is located at the bottom right of the form.

Figure 20: Raising an Issue Ticket in GARUDA Request Tracker

Once the problem is reported, the user gets a ticket number, which is used for further references. The user can also view the status of the ticket by logging in.

10. Frequently Asked Questions

GARUDA PORTAL

1. I am not able to invoke the Grid Portal using the above URL.
 - a. Check whether you have typed the correct URL
 - b. If the URL is verified to be correct and still the problem persists, try after 10 minutes. It should work. (The reasons may have been that the servers were undergoing temporary switchover).
 - c. If the problem still persists, then please report it using the GARUDA RT.
2. When do I use Advanced Submission?

When the application is Heterogeneous in nature and doesn't require the scheduler interface or need to be submitted directly to Globus.
3. I am not getting the expected outputs for the jobs that I submit.
 - a. Check the status of the cluster to which you submitted the job in the "Notices" option.
 - b. Check whether you have submitted an executable file, or have submitted a source file by mistake.
 - c. If you indeed have submitted an executable file, check whether the executable was compiled in the operating system to which it was submitted.
 - d. If you have submitted a heterogeneous job, the above two steps should be performed for each and every platform (operating system) on which you want to execute the job.
4. Do I need to learn Grid job submission languages?

No. GARUDA Portal users need not know Resource Specification Language (RSL) or Job Template (JT).

PARYAVEKSHANAM

1. When to use Paryavekshanam?

Paryavekshanam is used for troubleshooting or analysis of the problem when the user encounters problems while submitting the job to GARUDA Portal.

2. How can the Grid Monitoring Tool be invoked?

The GARUDA Grid monitoring Tool called Paryavekshanam can be invoked through any web browser or GARUDA Portal (best viewed through IE 6.0 and above) by using the following URL: <http://192.168.60.70/gridmon/GRID/gridmon.php>

3. Is a login required to use this tool?

No, login is required to use this tool.

4. What is the task of the Grid Operations Center (GOC)?

To monitor all the critical components of the GARUDA Grid on a 24X7 basis, and detect any malfunctioning. A first level fault analysis can be carried out and the concerned local center will be informed / alerted to rectify the problem immediately.

5. What are the links available on this page?

a. All the operational buttons of the locations in India map are linked to the pages corresponding to the details of the respective cities.

b. In the radar block, the text Grid Overview is linked to detailed overview of the graph.

6. What is radar graph and what it signifies?

It is also known as *star or spider graph*, is laid out in circular fashion. Radar graph consists of axis lines that start in the center of a circle and extend to its periphery. First axis is always vertical. Each axis represents parameter to be measured and they are expected to be positive in nature.

Radar graph is used to specify uniform utilization of the values. The ideal graph on plotting the values is a circle at 100%. It represents the parameter values in percent. Graph plotted represents the status of the utilization of resources.

7. How can the users contact for further information?

The users can send queries to gridmon@cdacb.ernet.in

INDIAN GRID CERTIFICATE AUTHORITY

1. Whom should I contact for IGCA certificate?

C-DAC, Knowledge Park,
#1, Old Madras Road, opp. Aero Engine Division, Bangalore, India - 560038
Phone: +91-80-25244059, Fax: +91-80-25247724
Email: igca@cdacb.ernet.in

2. I forgot the pass phrase of my user certificate. Can you please reset it to the default or call me with it or send it to me in an email ?

*NO. The pass phrase securing your private key is only stored/managed by you .
You must revoke your certificate and request a new one.*

3. My system got crashed Or got affected with virus and OS was installed, but I forget to export my certificate from browser . How should I do now ? Should I request the certificate again or re-write the application form ?

Yes, you should request a new user certificate. Keep the back up certificate onto a removable media which should be kept in safe.

4. How to check the expiration date of user/host certificate. If it will expire soon, what should I do ?

- a) Check the expiration date of the certificate:

\$ openssl x509 -in hostcert.pem -noout -enddate

\$ openssl x509 -in usercert.pem -noout -enddate

- b) *If the certificate is expiring, then follow the procedure to rekey.*

11. LIST OF REGIONAL AUTHORITIES (RA's)

Bangalore

| Organization | RA Name | Email | Phone |
|-------------------------|------------------|------------------------|------------------------------------|
| C-DAC, Knowledge Park | Divya .M .G | divyam@cdacb.ernet.in | +91-80-25244059 +91-09449994478 |
| C-DAC, Electronics City | Praveen Ampatt | praveen@ncb.ernet.in | +91-80-28523300 |
| SERC, IISc | Sathish Vadhiyar | vss@serc.iisc.ernet.in | +91-80-22932615 |

Pune

| Organization | RA Name | Email | Phone |
|-----------------------|---------------------|----------------------|-----------------|
| C-DAC | Akshara Kaginalkar | akshara@cdac.in | +91-20-25704226 |
| NCRA, Pune University | V Venkata Subramani | vvs@ncra.tifr.res.in | +91-20-25719202 |

Mumbai

| Organization | RA Name | Email | Phone |
|--------------|---------------|-------------------|------------------|
| BARC, Mumbai | Jagadeesh B S | jag@barc.gov.in | +91-022-25595214 |
| IIT, Mumbai | Petety Balaji | balaji@iitb.ac.in | +91-022-25767778 |

Chennai

| Organization | RA Name | Email | Phone |
|----------------|---------------------|----------------------------|-----------------|
| MIT | Kumar Rangasamy | rangasamykumarme@gmail.com | +91-44-22516015 |
| C-DAC, Chennai | Dhanesh .K .K | dhaneshkk@cdac.in | +91-44-22542226 |
| IMSC, Chennai | Mr. Raveendra Reddy | ravi@imsc.res.in | 044-22543222 |

Hyderabad

| Organization | RA Name | Email | Phone |
|--------------------|-------------------|----------------------|------------------------------------|
| C-DAC | Murty CH.A.S | chasmurty@cdac.in | +91-40-23150115 |
| Univ. of Hyderabad | E. A. Vinod Kumar | vinod@uohyd.ernet.in | +91-40-23138002 +91-09848415788 |

Delhi

| Organization | RA Name | Email | Phone |
|--------------|--------------|-------------------------|-------------------------------------|
| JNU | Andrew Lynn | andrew@mail.jnu.ac.in | +91-09968474785 |
| IGIB | Debasis Dash | ddash@igib.res.in | +91-011-27662738 |
| IIT, Delhi | Pragya Jain | pragya@cc.iitd.ernet.in | +91-09811304076 +91-011-26591783 |

Kolkata

| Organization | RA Name | Email | Phone |
|--------------|---------------|--------------------------|------------------|
| VECC | Vikas Singhal | vikas@veccal.ernet.in | +91-033-23182424 |
| VECC | Tapas Samanta | tsamanta@veccal.ernet.in | +91-033-23371230 |
| SAHA | Asit. K. De | asitk.de@saha.ac.in | +91-033-23375347 |

Ahmedabad

| Organization | RA Name | Email | Phone |
|--------------|-----------------|------------------|------------------|
| PRL | Jigarbhai Raval | jigar@prl.res.in | +91-079-26314035 |

Kharagpur

| Organization | RA Name | Email | Phone |
|----------------|-------------|----------------------------|-------------------------------------|
| IIT, Kharagpur | C. S. Kumar | kumar@mech.iitkgp.ernet.in | +91-03222-282934 +91-09434005819 |

Trivandrum

| Organization | RA Name | Email | Phone |
|--------------|-------------|-----------------------|-------------------------------------|
| VSSC | Gopakumar S | gopakumar@vssc.gov.in | +91-0471-2565889 +91-09496060095 |

Jammu

| Organization | RA Name | Email | Phone |
|---------------------|----------------|--------------------|-------------------|
| University of Jammu | Mr. Anik Gupta | anik.gupta@cern.ch | +91- 0191-2454389 |

Note: If there is no RA in your organization, please contact us directly - igca@cdacb.ernet.in.

12. REFERENCES

1. www.cdac.in
2. www.garudaindia.in
3. <http://ca.garudaindia.in/>
4. <http://192.168.60.40/GridPortal1.3/>
5. <http://192.168.60.70/gridmon/GRID/gridmon.php>
6. www.gridway.org
7. www.clusterresources.com
8. www.globus.org

Individual Product Manuals of the following GARUDA Components

1. GARUDA Portal
2. Gridhra
3. SRB

13. GLOSSARY (Alphabetically)

GLOBUS TOOLKIT

Open source Grid middleware for building computing grids, developed and provided by the Globus Alliance. It is designed to support the essential activities of a Grid namely Security, Execution management, Information Service, Data management and other common runtime components

GRIDWAY

It is a meta-scheduler which enables large-scale, reliable and efficient sharing of computing resources (clusters, computing farms, servers, supercomputers...), managed by different LRM (Local Resource Management) systems, such as PBS, SGE, LSF, Condor..., within a single organization (enterprise grid) or scattered across several administrative domains (partner or supply-chain grid).

IGCA

Indian Grid Certification Authority (IGCA) is the First CA in India established to address security issues of grid. The IGCA is the official body that addresses the key security concerns and requirements on interoperability between international grids. It issues digital certificate for users and grid resources including computers, which are recognized in the global grid community encompassing the International Grid Trust Federation (IGTF). Located at C-DAC Bangalore, the IGCA operates under the IGTF policy. The certificate policies and application procedures are explained in the website:

REQUEST TRACKER (RT)

RT is an enterprise-grade ticketing system which enables a group of people to intelligently and efficiently manage tasks, issues, and requests submitted by a community of users.

TORQUE

It is an open source resource manager providing control over batch jobs and distributed compute nodes.